

28. (New) A method as defined in claim 25 wherein the act of automatically identifying the one or more files to be analyzed comprises:

reviewing a my documents folder;
reviewing a recently used document list;
reviewing a list of marked documents for use offline;
reviewing a desktop folder; and
reviewing dependent files

REMARKS

This Amendment and Response is intended to fully respond to the Second Office Action dated November 21, 2002. In that Office Action, claims 1-6 and 8-20 were examined; claims 10, and 18-20 were rejected under 35 U.S.C. § 112 for insufficient antecedent basis as to one or more terms in the claims; and claims 1-6 and 8-20 were rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,995,756 to Herrmann.

Reconsideration of the rejections, in view of these remarks, is respectfully requested.

Claim Rejections - 35 U.S.C. § 112

Claims 10, 18, 19 and 20 were rejected under 35 U.S.C. § 112, second paragraph, for insufficient antecedent basis. Claims 10, 18, 19 and 20 have been amended to resolve any issue with regard to insufficient antecedent basis. As such, the Applicant earnestly requests removal of the rejection of claims 10, 18, 19 and 20 under 35 U.S.C. § 112.

Brief Description of the Invention

Before addressing the claim rejections based on the prior art, a brief description of the invention is hereby provided. The present invention relates generally to methods and systems for identifying files that a user of a client computer may need when the client computer is disconnected from a network. The files may be automatically or manually identified and may be related to file usage patterns. In order to execute the identified files on the client computer *when*

it is disconnected from the network, before the disconnection from the network occurs a handler associated with each file type identifies application software that may be required to execute each of the files. Each of the files (if not on the client computer already) and their associated application software is downloaded to the client computer. This is particularly helpful when most if not all the functionality typically occurs on a remote computer system, such as through the use of remote procedure calls, and very little of the application functionality has been installed on the client computer system. Such use of other network or remote computers to perform application functionality is an increasing phenomenon and therefore the problems associated with disconnecting a client computer system from the network have increased significance and hence, the motivation for the present invention.

Another particular problem that is solved by the present invention relates to the fact that some applications download or install functionality to a client computer “on demand.” That is, when connected to a network, a particular user may be working with a file along with some of its associated application, wherein that portion of the application functionality has been downloaded to the computer system. However, some of the other application functionality of that application has not been downloaded to the client computer system since the user, in this example, has not requested the other functional component(s). Upon requesting the functionality, then and only then would that functionality be downloaded/transferred to the client computer system and installed, i.e., the functionality is installed on demand. (See Specification, p. 8, l. 34 through p. 9, l. 3). In this example, if the user disconnects from the network before requesting such functionality, then that functionality will not be available while the computer is disconnected from the network. The present invention solves this problem by identifying possible functionality that may be needed while working with the file offline and transfers such functionality to the client computer system before the computer is disconnected from the network. Of course, other problems are also solved using the present invention, but this particular problem highlights the nature of the invention.

Claim Rejections - 35 U.S.C. § 103

Claims 1-6 and 8-20 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,995,756 to Herrmann. In particular, the Examiner relied on Herrmann as teaching all the claim elements of each claim. Importantly, the Examiner noted that the Herrmann is silent with respect to running the files when the computer is disconnected, but relies on the assumption that one skilled in the art would deem it obvious to improve the implementation of Herrmann to include running the set of files when disconnected since the client computer in Herrmann is capable of running MIME type files using the downloaded application offline or independent of the web server.

REBUTTAL OF PRIMA FACIE CASE OF OBVIOUSNESS

Applicant respectfully traverses the section 103 rejections. The Examiner has failed to substantiate a prima facie case of obviousness because one of the requirements of a prima facie case is absent. In particular, the cited reference (only one reference was cited) does not teach or suggest all the limitations of independent claims 1, 4, or 10 (or new independent claim 21). Specifically, Herrmann does not disclose, teach or suggest the limitation related identifying application functionality needed to run when the computer is disconnected from the network and, in the case of claims 4, 10 and 21 the limitation of transferring and/or installing application functionality to the client computer system not previously transferred so that the application functionality is available when the client computer system is disconnected from the network.

Specifically, Herrmann describes a system and method for executing an application located on a remote computer system, wherein the application has been divided into frames, and wherein each frame has its own functionality. (See Col. 7, ll. 49-60 of Herrmann) As best understood, the user essentially operates his/her browser to access specific frames of the application. When a new frame is requested, an HTTP request is invoked to get the proper functionality, which comes in the form of another MIME file. The MIME file includes information as to how to download the operational code, and such downloading occurs shortly after receiving the MIME file. Herrmann states "When the user selects a hyperlink that points to an application page, the Web browser downloads the application code and executes the page inside the browser, as FIG. 4C illustrates." (Col. 9, ll. 38-41) The execution involves the

browser invoking a handler to check locally for the application code and then, if not found locally, downloading the code. However, not all components are downloaded at the same time. Herrmann states:

“When the code is downloaded, a ‘code base’ specification (file) is initially requested from the server. The code base itself can range from a simple DLL file to a Cabinet file (Microsoft .cab file) containing multiple compressed files. Still further, an information (e.g., Microsoft .inf) file can be employed for instructing the client system how to install the downloaded application. These mechanisms afford great flexibility in choosing which component of an application gets downloaded and when.” (Herrmann, Col. 9, ll. 57-65, emphasis added here).

As a result, Herrmann provides a system of accessing code on an “on-demand” basis. This is similar to the on-demand application described in the specification of the present application wherein only portions of the functionality are downloaded at a time.

Contrary to the pending claims however, Herrmann does not teach or suggest a system or method that identifies application functionality to be used when the computer is disconnected from the network, independent from previously requested functionality. Herrmann specifically does not relate to identifying functionality based on an analysis of files and the potential functionality associated with or related to the files, let alone transferring and installing such functionality to and on the client computer system for use when the computer is disconnected from the network.

Importantly, Herrmann does not even consider the client as becoming disconnected from the network. Based on the teaching of Herrmann, only code that is to be used in accordance with a specific request should be downloaded. Indeed, Herrmann states: “Once the application is executing at the client, such as illustrated in FIG. 4D, it can execute remote logic such as using RPC (Remote Procedure Call) methodology. In this manner business logic which is preferably implemented as remote procedures can still be used, as before.” (Herrmann, Col. 10, ll. 20-25). Clearly, Herrmann does not contemplate transferring all the functionality to the client computer system and most importantly, Herrmann does not contemplate transferring functionality such that the client computer system can operate offline. Moreover, Herrmann does not suggest the desirability of the present invention. (See MPEP 2143.01 “The prior art must suggest the desirability of the claimed invention.”)

If anything, Herrmann teaches away from the present invention in that it relies on user connectivity with the network to provide application information while maintaining the remote procedure call, or “business logic” methodology in place. Moreover, Herrmann further states:

“Once the form is created, it can establish connections back to any remote server objects it needs to perform its functions. At this point, the user can interact with the form, which will appear embedded in the Internet Explorer frame. When the user changes to a different page, the browser assume [sic] responsibility for eventually closing and destroying the form (and relinquishing any outstanding connection to the remote servers.) (Herrmann, col. 9, ll. 4-11).

It is clear that Herrmann only contemplates the use of any downloaded functionality as part of a “online” session and thus does not suggest the use of such functionality offline. Herrmann does not suggest the analysis of a file to determine if some functionality should be downloaded such that the functionality is available once the computer is disconnected from the network. The Examiner has failed to point to any teaching or suggestion in Herrmann that would teach or suggest the analysis of a file to then download functionality such that the functionality would be available offline, i.e., for the purpose of operating offline.

USE OF IMPERMISSIBLE HINDSIGHT

“To imbue one of ordinary skill in the art with knowledge of the invention in suit, when no prior art reference or references of record convey or suggest that knowledge, is to fall victim to the insidious effect of a hindsight syndrome wherein that which only the inventor taught is used against its teacher.” W.L. Gore, 721 F.2d at 1553, 220 USPQ at 312-13.

A prima facie case of obviousness cannot be established from the use of impermissible hindsight. The Examiner has not shown any suggestion, either in the references or in the knowledge of one skilled in the art, that the language of Herrmann could be interpreted to include the analysis of a file to then download functionality such that the functionality would be available offline when the computer is disconnected from the network. **“It is impermissible within the framework of section 103 to pick and choose from any one reference only so much of it as will support a given position to the exclusion of other parts necessary to the full appreciation of what such reference fairly suggests to one skilled in the art.”** Bausch &

Lomb, Inc. v. Barnes-Hind/Hydrocurve, Inc. 796 F.2d 443, 230 USPQ 416 (Fed. Cir. 1986). See also In re Wesslau, 353 F.2d 238, 241, 147 USPQ 391, 393 (CCPA 1965); In re Mercer, 515 F.2d 1161, 1165-66, 185 USPQ 774, 778 (CCPA 1975). **“There must be a teaching or suggestion within the prior art, or within the general knowledge of a person of ordinary skill in the field of the invention, to look to particular sources of information, to select particular elements, and to combine them in the way they were combined by the inventor.”** ATD Corp. v. Lydall, Inc., 159 F.3d 534, 48 USPQ2d 1321 (Fed. Cir. 1998).

Since Herrmann does not teach or suggest the use of functionality upon disconnecting a computer from the network without the use of impermissible hindsight, such a reference provides no basis for a rejection under 35 U.S.C. 103, as a matter of law. For this reason, independent claims 1, 4, 10 and 21 are believed to be allowable. Moreover, since each of the remaining pending claims depend from one of the independent claims, such dependent claims are believed to be allowable for at least this reason.

Conclusion

As originally filed, the present application included 11 claims, 3 of which were independent. Following the amendment filed on July 25, 2002, (in which one claim was canceled and nine new claims were added) the application included 19 pending claims, 3 of which were independent. Following this amendment (in which no claims were canceled and 8 new claims were added), the application now includes 27 pending claims, 4 of which are independent. Accordingly, a check for \$210.00, for the one additional independent claim exceeding 3 independent claims and the 7 additional claims exceeding 20 total claims is submitted herewith. It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the above remarks and amendments, it is believed that the application is now in condition for allowance, and such action is respectfully requested. Claims 1-6 and 8-28 remain pending in the application and are believed to clearly be allowable over the art of record. A separate markup with the claim amendments shown by bracketing and underlining is enclosed with this amendment in accordance with 37 C.F.R. 1.121. Should the Examiner have any remaining questions or concerns, he is encouraged to contact the undersigned attorney by telephone to expeditiously resolve such concerns.

Respectfully submitted,

Dated: 2/21/03





Timothy B. Scull, Attorney Reg. No. 42,137
MERCHANT & GOULD P.C.
P. O. Box 2903
Minneapolis, MN 55402-0903
(303) 357-1648

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Belt et al.	Examiner:	Anya, Charles E.
Serial No.:	09/223,595	Group Art Unit:	2151
Filed:	December 30, 1998	Docket No.:	36711.1 60001.99US01
Title:	METHOD AND SYSTEM FOR IDENTIFYING A PROGRAM MODULE FUNCTIONALLY NEEDED BY A COMPUTER WHEN DISCONNECTED FROM A NETWORK		

AMENDMENTS -- MARKUP**IN THE SPECIFICATION:**

Please amend the paragraph beginning on page 7, line 22 and ending on page 7, line 30 as shown in the following paragraph:

Before discussing Fig. 2, a brief discussion of terminology is needed. In accordance with an exemplary embodiment of the present invention, the installer program module **37** recognizes three principal elements: products, features and components. The installer program module **37** is also described in co-pending application serial no. [] 09/158,125 entitled "Use of Relational Databases for Software Installation", which is assigned to the same assignee, filed on September 21, 1998, and incorporated by reference herein.

IN THE CLAIMS:

Please amend the claims 10 and 18-20 and add new claims 21- 28 as set forth below:

10. (Amended) A method for identifying a set of application functionality to be stored on a computer connected to a network, comprising the steps of:
- causing a document identification engine (DIE) to create a list of a plurality of files stored locally on the computer;
 - sending the list of files from the DIE to a document mapping engine (DME);
 - causing the DME to identify a proper handler routine for each file in the list of

files;

sending each file from the DME to the proper handler routine;

causing the handler routine to identify the application functionality needed to execute each file when the computer is disconnected from the network;

sending a list of needed application [program] functionality of the handler routine to the DME;

sending a list of needed application [program] functionality from the DME to a migration engine (ME);

causing the ME to determine the current status of the needed application functionality; and

if the status of the needed application functionality indicates that the needed application functionality is not installed locally on the computer, then causing the ME to install the needed application functionality to the computer.

18. (Amended) The method of claim 10 further comprising the steps of:

causing the handler routine to notify the DME of an embedded file; and

in response to receiving the notification of the embedded file, causing the DME to transmit the embedded file to another handler routine associated with the embedded file.

19. (Amended) The method of claim 10 further comprising the steps of:

sorting the application [program] functionality according to a frequency of occurrence.

20. (Amended) The method of claim 19 wherein the step of sorting the application [program] functionality comprises steps of:

causing the handler routine to return importance rankings associated with the application [program] functionality.

21. (New) A method of executing application functionality on a computer disconnected from a network comprising:

while the computer is connected to the network, identifying one or more files to be used when the computer is disconnected from the network;

determining whether the one or more files are associated with the application functionality, wherein the application functionality is located on the network;

upon identifying one or more files associated with the application functionality, transferring the application functionality from the network to the computer;

disconnecting the computer from the network; and

executing the application functionality in combination with at least one of the files associated with the application functionality.

22. (New) A method as defined in claim 21 wherein the one or more files are on the computer prior to the identifying act.

23. (New) A method as defined in claim 21 wherein the one or more files are on the network prior to the identifying act, the method further comprising:

prior to disconnecting the computer from the network, transferring the one or more files from the network to the computer.

24. (New) A method as defined in claim 21 wherein the act of determining whether the one or more files are associated with the application functionality comprises analyzing the one or more file to determine application functionality

25. (New) A method as defined in claim 24 further comprising:

receiving a disconnect signal; and

in response to the disconnect signal, automatically identifying the one or more files to be analyzed.

26. (New) A method as defined in claim 25 wherein the disconnect signal comprises a shut-down signal.

27. (New) A method as defined in claim 25 wherein the disconnect signal comprises a undock signal.

28. (New) A method as defined in claim 25 wherein the act of automatically identifying the one or more files to be analyzed comprises:

reviewing a my documents folder;

reviewing a recently used document list;

reviewing a list of marked documents for use offline;

reviewing a desktop folder; and

reviewing dependent files.